



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/757,517	01/10/2001	Stephen E. Fischer	FIS920000280US1	8455

29505 7590 08/24/2004

DELIO & PETERSON, LLC
121 WHITNEY AVENUE
NEW HAVEN, CT 06510

EXAMINER

FOWLKES, ANDRE R

ART UNIT	PAPER NUMBER
----------	--------------

2122

DATE MAILED: 08/24/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Advisory Action

Application No.

09/757,517

Applicant(s)

FISCHER, STEPHEN E. 

Examiner

Andre R. Fowlkes

Art Unit

2122

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

THE REPLY FILED 19 July 2004 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE. Therefore, further action by the applicant is required to avoid abandonment of this application. A proper reply to a final rejection under 37 CFR 1.113 may only be either: (1) a timely filed amendment which places the application in condition for allowance; (2) a timely filed Notice of Appeal (with appeal fee); or (3) a timely filed Request for Continued Examination (RCE) in compliance with 37 CFR 1.114.

PERIOD FOR REPLY [check either a) or b)]

- a) ☐ The period for reply expires _____ months from the mailing date of the final rejection.
- b) ☒ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection. **ONLY CHECK THIS BOX WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).**

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

1. ☐ A Notice of Appeal was filed on _____. Appellant's Brief must be filed within the period set forth in 37 CFR 1.192(a), or any extension thereof (37 CFR 1.191(d)), to avoid dismissal of the appeal.
2. ☐ The proposed amendment(s) will not be entered because:
- (a) ☐ they raise new issues that would require further consideration and/or search (see NOTE below);
 - (b) ☐ they raise the issue of new matter (see Note below);
 - (c) ☐ they are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or
 - (d) ☐ they present additional claims without canceling a corresponding number of finally rejected claims.

NOTE: _____.

3. ☐ Applicant's reply has overcome the following rejection(s): _____.
4. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).
5. ☒ The a) ☐ affidavit, b) ☐ exhibit, or c) ☒ request for reconsideration has been considered but does NOT place the application in condition for allowance because: See Continuation Sheet.
6. ☐ The affidavit or exhibit will NOT be considered because it is not directed SOLELY to issues which were newly raised by the Examiner in the final rejection.
7. ☒ For purposes of Appeal, the proposed amendment(s) a) ☐ will not be entered or b) ☒ will be entered ~~and an explanation of how the new or amended claims would be rejected is provided below or appended.~~

The status of the claim(s) is (or will be) as follows:

Claim(s) allowed: _____.

Claim(s) objected to: _____.

Claim(s) rejected: 1-7, 9-22 & 24-26.

Claim(s) withdrawn from consideration: _____.

8. ☐ The drawing correction filed on _____ is a) ☐ approved or b) ☐ disapproved by the Examiner.
9. ☐ Note the attached Information Disclosure Statement(s) (PTO-1449) Paper No(s). _____.
10. ☒ Other: See Continuation Sheet

Continuation of 5. does NOT place the application in condition for allowance because: applicants arguments do not distinguish the application over the prior art (see item 10).

Continuation of 10. Other: As per the arguments presented on pages 11-23 of the after final amendment, the Examiner disagrees with the applicant's characterization of the applied art and contends that the applicant's amendments and arguments do not distinguish the application over the prior art for the following reasons.

The applicant has argued that the GNU Make Manual, (GNU MM), does not disclose reading a plurality of dependency files associated with target files into a single control file and splitting such dependency files into target strings and prerequisite strings, on p. 16:10-15. Rather, the pages of The GNU Make Manual cited by the Examiner (GNU MM, p.14, 11. 9-11) merely disclose that when the objects of a make file are created only by implicit rules, entries may be grouped by their prerequisites instead of by their targets.

The examiner disagrees with the applicant's characterization of the GNU MM reference and further, contends that the GNU MM grouping of the entries by their prerequisites allows for the identical operations/functions to be performed on the prerequisites and target strings as in the situation where the target and prerequisite strings are separated, as noted on p. 4 line 1 – p. 5 line 10 of the final rejection. Since, identical functions are capable of being performed, the separation of the dependency files into target and prerequisite strings and the use of a single control file are a matter of design choice and do not distinguish the instant application over the prior art.

The applicant has argued that the GNU Make Manual does not recite performing programming language substitutions in these target strings using an interpreted scripting language, whereby target strings having programming language substitutions are appended to a requisition list while the prerequisite strings are stored in corresponding requisite arrays, on p. 16:19-17:1.

The examiner contends that the application, as claimed, is not distinguishable over the prior art. The GNU MM discloses that programming substitutions are performed (p. 16 lines 4-5, "a text string value ... can be substituted"), the use of an interpreted scripting language (p. 97 line 45, the use of a "command script" and p. 47 lines 40-44, "commands in make files are always interpreted"). Additionally, strings are appended to a list and stored on (p. 18 lines 3, "appending (strings)").

The applicant has argued that the GNU Make Manual does not disclose updating the split target files if it is determined that the prerequisite strings in the control file have been updated more recently than the substituted target string (re: claim 5), on p.17:15-17 of the after final amendment; however, the claims are not directed to updating the split target files.

The applicant has argued that the GNU Make Manual does not disclose performing such a step by executing an algorithm that matches target files from the target file list with the substituted target string in the requisition list (claims 1 and 16) if it is determined that the corresponding prerequisite strings, have been updated more recently than the substituted target string, on p. 17:17-22.

The examiner contends that this limitation does not further distinguish the application over the prior art. The GNU MM discloses executing an algorithm that matches files with strings, (p. 38 lines 28-29, "(when) the string ... match(es) the file name"), and the ability to determine which files have been updated most recently (p. 6 lines 39-41, "The make program uses the ... last –modification times of the files").

The applicant has argued, on p. 19:4-13, that with respect to new independent claim 26, that The GNU Make Manual's Gmake's vpath/VPATH facility does not disclose multi-directory builds from a single control file. Further, the passages of The GNU Make Manual cited by the Examiner, namely, "the directory search features of make facilitate this by searching: several directories automatically to find a prerequisite" at page 28, lines 51-53, do not disclose update rules that support multi-directory builds from a single control file.

The examiner contends that rules supporting multi-directory builds are supported by the GNU MM, (p. 33 lines 39-43, "When a prerequisite's name has the form -lname, make handles it specially by searching for the file libname.so in the current directory, in directories specified by matching vpath search paths and the VPATH search path, and then in the directories /lib, /usr/lib, and prefix/lib", and a single file can contain this prerequisite, requiring multi-directory builds).

The applicant has argued, on p. 20:3-4, that the directory searching in The GNU Make Manual's gmake is not dynamic directory switching to specify multiple files in multiple directories as is recited in claim 24. Applicant's dynamic directory switching allows the user to conveniently switch from one directory to another as he or she specifies prerequisites.

The Examiner contends that the GNU MM anticipates specifying prerequisites to allow the dynamic directory switching described in the instant application. The GNU MM discloses (p. 33 lines 39-43, "When a prerequisite's name has the form -lname, make handles it specially by searching for the file libname.so in the current directory, in directories specified by matching vpath search paths and the VPATH search path, and then in the directories /lib, /usr/lib, and prefix/lib" and several of such prerequisites, indicating different directories, may be specified).

The applicant has argued, on p. 20:7-10, that w.r.t. claim 25, the GNU Make Manual does not disclose that an algorithm considers a directory to be out-of-date regardless of its time stamp (i.e., always out-of-date regardless of its time stamp) such that any rule associated with directory target is always triggered.


The Examiner contends that the GNU Make Manual does disclose the limitations of claim 25. Using the directory part of the file name, as shown on p. 124:25-28, to select all files within the directory and (p. 21:6-7, "use a match-anything pattern rule to say that to remake any target (in the directory)").

The applicant argues, on p. 22:10-23:7, that the Welch reference does not overcome the above deficiencies of the GNU Make Manual. Welch is merely directed to Tcl scripting and is delivered as a practical guide to help users of the Tcl and Tk programming languages get the most out of Tcl and Tk. The Welch reference does not disclose reading a plurality of dependency files associated with target files into a control file and splitting such dependency files into target strings and prerequisite strings. As such, Welch does not disclose performing programming language substitutions in these target strings using an interpreted scripting language, and executing a utility program to

update the split target files if it is determined that the prerequisite strings in the control file have been updated more recently than the substituted target string (claims 1, 5 & 16). The applicant argues further that neither The GNU Make Manual nor the Welch reference, taken singly or in any proper combination thereof, discloses the instant invention, such that, pending claims 12, 13 and 20-22 are neither anticipated by nor rendered obvious over The GNU Make Manual or the Welch reference, alone or in any proper combination.

The Examiner disagrees with the applicant's characterization of the applied references. The Welch reference is not cited to address the limitations of claims 1, 5 & 16, as indicated by the applicant on p. 22:16-23:1 of the after final amendment. The GNU MM is cited to address the limitations of claims 1, 5, & 16, while the GNU MM/Welsh combination is used to address the additional limitations imposed by claims 12, 13 and 20-22. The examiner contends that the Welch reference is cited to disclose that one skilled in the art, at the time the invention was made, would have had motive to combine the teachings of the Welch reference with the system of the GNU MM. One would have been motivated to combine Tcl with the GNU MM, to satisfy the limitations of claims 12, 13 and 20-22, because Tcl is a useful language that allows one to build complex programming scripts, yet it is easy to combine with current applications (Final action p. 11:1-3).

For the above stated reasons, the after final amendment does not place the application in condition for allowance.



TUAN DAM
SUPERVISORY PATENT EXAMINER